Machine Learning

Dmytro Fishman (dmytro.fishman@ut.ee)

UNIVERSITY of TARTU
Institute of Computer Science
<table>
<thead>
<tr>
<th>Assignment</th>
<th>Date of assignment</th>
<th>Deadline (midnight 23:59)</th>
</tr>
</thead>
<tbody>
<tr>
<td>HW1</td>
<td>Sep 7</td>
<td>Sep 20</td>
</tr>
<tr>
<td>HW2</td>
<td>Sep 21</td>
<td>Oct 4</td>
</tr>
<tr>
<td>HW3</td>
<td>Oct 5</td>
<td>Oct 18</td>
</tr>
<tr>
<td>Paper summary</td>
<td>Oct 12</td>
<td>Oct 26</td>
</tr>
<tr>
<td>HW4</td>
<td>Oct 19</td>
<td>Nov 1</td>
</tr>
<tr>
<td>HW5</td>
<td>Nov 9</td>
<td>Nov 22</td>
</tr>
<tr>
<td>HW6</td>
<td>Nov 23</td>
<td>Dec 6</td>
</tr>
<tr>
<td>Project</td>
<td>TBA</td>
<td>Dec 14</td>
</tr>
</tbody>
</table>
Recap
Machine Learning

- Supervised Learning
- Unsupervised Learning
- Reinforcement Learning
Unsupervised Learning

Supervised Learning

Reinforcement Learning

Classification

Regression
Machine Learning

- Classification
- Regression
- Supervised Learning
- Reinforcement Learning

Unsupervised Learning
- Clustering

Dimensionality Reduction
Dimensionality reduction

Unsupervised Learning

High Dimensional Space

Low Dimensional Space

Torus image credit: https://mathematica.stackexchange.com/questions/39879/create-a-torus-with-a-hexagonal-mesh-for-3d-printing
What is the problem with high-dimensional things?

- Hard to visualise
- Algorithms tend to get slow
- Methods trained on high-dimensional data suffer from the curse of dimensionality
What is the **curse of dimensionality**?

In order to keep high-dimensional space reasonably covered you need a lot more data.

On average **55.5%** of cells will be either **empty** or **singletons**

On average **92.5%** of cells will be either **empty** or **singletons**
200-Dimensional data?

Are all of these dimensions *equally useful*?
Principle components are **not additional axes/dimensions**

**How many** PCs will be formed in **200D** space?

**First few PCs** would be **enough** to capture important information
Compute covariance matrix

Perform eigen decomposition

Compute new coordinates
Supervised Learning pipeline

1. 200D raw data
2. Normalisation (subtract mean)
3. Train/test split
4. PCA 200 PCs
5. Find the best model using CV
6. Evaluate final model on the test set

Keep few PCs (90% variance)
Safe place
Test
Profit
Figure 6: A comparison between UMAP and t-SNE projections of a 3D woolly mammoth skeleton (50,000 points) into 2 dimensions, with various settings for parameters. Notice how much more global structure is preserved with UMAP, particularly with larger values of `n_neighbors`.

UMAP explained and compared to t-SNE: https://pair-code.github.io/understanding-umap/
Machine Learning

- Classification
- Regression
- Supervised Learning
- Reinforcement Learning

Dimensionality reduction

- Unsupervised Learning
- Clustering
Dimensionality reduction

Unsupervised Learning

Clustering

Unlabelled data
Clustering methods are looking for structure in unlabelled data.
How come stuff does not have annotations?
(Why unsupervised learning is a thing?)
How come stuff does not have annotations?
(Why unsupervised learning is a thing?)

Labels are noisy and not trustworthy
How come stuff does not have annotations?
(Why unsupervised learning is a thing?)

Labels are noisy and not trustworthy

Labels are super expensive and laborious
How come stuff does not have annotations?
(Why unsupervised learning is a thing?)

Labels are noisy and not trustworthy

Labels are super expensive and laborious

Labels simply do not exist
How come stuff does not have annotations?
(Why unsupervised learning is a thing?)

Labels are noisy and not trustworthy

Labels are super expensive and laborious

Labels simply do not exist

We learn with very limited or no supervision
Two groups labeled by an expert
No labels. Are there any meaningful groups?
Hierarchical clustering

K-means

DBSCAN

noise

centroids
Hierarchical clustering
Hierarchical clustering

At first all points are considered to be separate clusters.
Hierarchical clustering

Two **closest** clusters are **merged** into one
Hierarchical clustering

Two closest clusters are merged into one

Euclidean distance is calculated between all clusters
Hierarchical clustering

Two closest clusters are merged into one

Euclidean distance is calculated between all clusters
Hierarchical clustering

Two closest clusters are merged into one

Euclidean distance is calculated between all clusters
Hierarchical clustering

Two *closest* clusters are *merged* into one

These points are now belong to one *cluster*
Hierarchical clustering

Two **closest** clusters are **merged** into one

**Euclidean** distance is calculated between all clusters
Hierarchical clustering

Now we have to **re-calculate distances** between clusters again to figure out next pair of **clusters to merge**.
Hierarchical clustering

When distance is calculated between single point cluster and multiple point cluster few options exist.
Hierarchical clustering

When distance is calculated between single point cluster and multiple point cluster few options exist

1. Distance between closest points defines the distance between clusters (i.e. single linkage)
Hierarchical clustering

When distance is calculated between **single point** cluster and **multiple point cluster** few options exist

I. Distance between **closest points** defines the distance between clusters (i.e. **single linkage**)

II. Distance between **furthest points** defines the distance between clusters (i.e. **complete linkage**)

---

**Note:**

- **Single point cluster** refers to a cluster containing a single data point.
- **Multiple point cluster** refers to a cluster containing multiple data points.
- **Distance** is a measure of similarity or dissimilarity between clusters.
- **Single linkage** and **complete linkage** are methods used to calculate the distance between clusters.
Hierarchical clustering

When distance is calculated between **single point** cluster and **multiple point cluster** few options exist.

I. **Distance** between **closest points** defines the distance between clusters (i.e. **single linkage**).

II. **Distance** between **furthest points** defines the distance between clusters (i.e. **complete linkage**).

III. **Distance** between **averages** defines the distance between clusters (i.e. **average linkage**).
Hierarchical clustering

Two **closest** clusters are **merged** into one

**Euclidean** distance is calculated between all clusters

*assume we use **single linkage***
Hierarchical clustering

Two closest clusters are merged into one

Euclidean distance is calculated between all clusters

*assume we use single linkage
Hierarchical clustering

Two closest clusters are merged into one

Euclidean distance is calculated between all clusters

*assume we use single linkage
Hierarchical clustering

When distances are equal, merge a pair at random

Euclidean distance is calculated between all clusters

Two closest clusters are merged into one

*assume we use single linkage
Hierarchical clustering

Two closest clusters are merged into one

Euclidean distance is calculated between all clusters

*assume we use single linkage
Hierarchical clustering

Two **closest** clusters are **merged** into one

Euclidean distance is calculated between all clusters

*assume we use **single linkage**
Hierarchical clustering

Two closest clusters are merged into one

Euclidean distance is calculated between all clusters

*assume we use single linkage
Hierarchical clustering

Two **closest** clusters are **merged** into one

**Euclidean** distance is calculated between all clusters

*assume we use **single linkage**
Hierarchical clustering

Two closest clusters are merged into one

Euclidean distance is calculated between all clusters

*assume we use single linkage
Hierarchical clustering

Two closest clusters are merged into one

Euclidean distance is calculated between all clusters

*assume we use single linkage
Hierarchical clustering

Two **closest** clusters are **merged** into one

**Euclidean** distance is calculated between all clusters

*assume we use **single linkage***
Hierarchical clustering

Two **closest** clusters are **merged** into one

Euclidean distance is calculated between all clusters

*assume we use **single linkage**
Hierarchical clustering

Two closest clusters are merged into one

Euclidean distance is calculated between all clusters

*assume we use single linkage
Hierarchical clustering

Two **closest** clusters are **merged** into one

**Euclidean** distance is calculated between all clusters

*assume we use **single linkage**
Hierarchical clustering
Hierarchical clustering

Diagram showing clusters and distance.
Hierarchical clustering

Dendrogram

Distance

Clusters
Hierarchical clustering
Hierarchical clustering

Distance

Clusters

Dendrogram
Hierarchical clustering

Distance

Clusters

Dendrogram
Hierarchical clustering

Distance

Clusters

Dendrogram
Hierarchical clustering

Distance

Clusters

Dendrogram
Hierarchical clustering
Hierarchical clustering
Hierarchical clustering

Distance

Clusters

Dendrogram

Dendrogram
Hierarchical clustering

Why are they on the same height?
Hierarchical clustering

Why are they on the same height?

Because **distances** between clusters are the same!
Hierarchical clustering
Hierarchical clustering
Hierarchical clustering

These are also the same!
Hierarchical clustering

Distance

Clusters

Dendrogram
Hierarchical clustering

Distance

Clusters

Dendrogram

2 0 1 2 3 4 5 6
Hierarchical clustering

Distance

Clusters

Dendrogram
Hierarchical clustering

Dendrogram
Hierarchical clustering

What is it useful for?
Hierarchical clustering

What is it useful for?

We can **cut it**!
Hierarchical clustering

We can **cut it!**
Give me **two** clusters!

What is it useful for?
Hierarchical clustering

We can **cut it**!
Give me **two** clusters!

What is it useful for?
Hierarchical clustering

What is it useful for?

We can cut it!
Give me two clusters!

Here are your two clusters

What is it useful for?
Hierarchical clustering

We can cut it!
Give me three clusters!

What is it useful for?
Hierarchical clustering

We can cut it!
Give me three clusters!

What is it useful for?
Hierarchical clustering

We can **cut it!**
Give me **three** clusters!

What is it useful for?
Hierarchical clustering

We can cut it!
Give me three clusters!

Here are your three clusters

What is it useful for?
Hierarchical clustering

Dendrogram is a way to store a structure of our data

What is it useful for?
Hierarchical clustering

K-means

centroids

DBSCAN

noise
K-means clustering
K-means clustering

Although they do have something in common with K-nearest neighbour, they are not the same.
All points are considered to be clusterless at first.
K-means clustering

Randomly spawn $K$ potential cluster centres
K-means clustering

Randomly spawn $K$ potential cluster centres (multiple possibilities)
K-means clustering

Randomly spawn K potential cluster centres (multiple possibilities)
K-means clustering

Randomly spawn \( K \) potential *cluster centres* (multiple possibilities)

Choose centroids *randomly within* dataset ranges
K-means clustering

Randomly spawn \( K \) potential cluster centres (multiple possibilities)
K-means clustering

Randomly spawn \( K \) potential cluster centres (multiple possibilities)

Random cases from dataset are selected as centroids
K-means clustering

Random cases from dataset are selected as centroids

Randomly spawn \( K \) potential cluster centres (multiple possibilities)
K-means clustering

Randomly spawn \( K \) potential cluster centres (multiple possibilities)

There are more options… (methods of initializing K-means clustering)

https://stats.stackexchange.com/questions/317493/methods-of-initializing-k-means-clustering
K-means clustering

Randomly spawn $K$ potential cluster centres (multiple possibilities)

Choose centroids randomly within dataset ranges
Randomly spawn **K** potential **cluster centres** (multiple possibilities)

**New centroids** (cluster centres)
Now as centroids were assigned we can start iterating K-means algorithm.
K-means clustering

Compute **distances** from all points to new centroids
K-means clustering

Assign each point to the closest centroid

Compute distances from all points to new centroids
K-means clustering

Assign each point to the closest centroid

Compute distances from all points to new centroids
K-means clustering

Assign each point to the closest centroid

Compute distances from all points to new centroids
K-means clustering

1. Compute distances from all points to new centroids.
2. Assign each point to the closest centroid.
K-means clustering

Compute distances from all points to new centroids

Assign each point to the closest centroid

Break the tie by random choice
**K-means clustering**

Compute distances from all points to new centroids.

Assign each point to the closest centroid.

Break the tie by random choice.
K-means clustering

1. Assign each point to the closest centroid.
2. Compute distances from all points to new centroids.
K-means clustering

Assign each point to the closest centroid

Compute distances from all points to new centroids
K-means clustering
K-means clustering

These are our clusters so far…

Cluster #1

Cluster #2
K-means clustering

Recalculate **centroids** for each cluster
K-means clustering

Recalculate centroids for each cluster
K-means clustering

Recalculate centroids for each cluster

\[ [2, 3, 5] \]

\[ \bar{x} = ? \]
K-means clustering

Recalculate centroids for each cluster

$\bar{x} = 3.33$
Recalculate centroids for each cluster.

\[\bar{x} = 3.33\]

\[\bar{y} = ?\]
K-means clustering

Recalculate centroids for each cluster

\[ \bar{x} = 3.33 \quad \bar{y} = 6 \]
K-means clustering

Recalculate centroids for each cluster

\[
\bar{x} = 3.33 \quad \bar{y} = 6
\]

New centroid for blue cluster

Recalculate centroids for each cluster
K-means clustering

Recalculate centroids for each cluster

\[ [3, 6, 7, 7] \]

\[ \bar{x} = 5.75 \]
K-means clustering

Recalculate centroids for each cluster

[3, 6, 7, 7]  [4, 6, 6, 8]

\[ \bar{x} = 5.75 \quad \bar{y} = 6 \]
K-means clustering

Recalculate centroids for each cluster

[3, 6, 7, 7]  \[4, 6, 6, 8\]

\(x = 5.75\)  \(y = 6\)
K-means clustering

Recalculate centroids for each cluster

$[3, 6, 7, 7] \quad [4, 6, 6, 8]$

$\bar{x} = 5.75 \quad \bar{y} = 6$
Now as centroids were assigned we can start iterating K-means algorithm.
K-means clustering

Compute **distances** from all points to new centroids
K-means clustering
K-means clustering

These are our clusters so far...
Recalculate **centroids** for each cluster.
K-means clustering

Recalculate centroids for each cluster
Recalculate **centroids** for each cluster.

K-means clustering
K-means clustering

Rinse and repeat until centroids are stable within certain radius (epsilon)
K-means clustering

Rinse and repeat until centroids are stable within certain radius (epsilon)
K-means clustering

Rinse and repeat until centroids *are stable* within certain radius (epsilon)
K-means clustering

Rinse and repeat until centroids are stable within certain radius (epsilon)
K-means clustering

Rinse and repeat until centroids are stable within certain radius (epsilon)
K-means clustering

K-means has been shown to converge in finite number of steps

Online visualisation: https://www.naftaliharris.com/blog/visualizing-k-means-clustering/
In both algorithms $K$ plays an important role
Any good way to choose $K$?
Any good way to choose $K$?

The rule of thumb is to choose $\sqrt{n/2}$ as $K$. 
Any good way to choose $K$?

The rule of **thumb** is to choose $\sqrt{\frac{n}{2}}$ as $K$

**Elbow method**: increase $K$ until it does not help to describe data better
Elbow method to predict K

K = 1

centroid

Elbow method for K-means: https://bl.ocks.org/rpgove/0060ff3b656618e9136b
Elbow method to predict K

K = 1

K = 2

Elbow method for K-means: https://bl.ocks.org/rpgove/0060ff3b656618e9136b
Elbow method to predict K

Elbow method for K-means: https://bl.ocks.org/rpgove/0060ff3b656618e9136b
Elbow method to predict $K$

For each case we calculate \textbf{sum of squared errors} (SSE)

Elbow method for K-means: https://bl.ocks.org/rpgove/0060ff3b656618e9136b
Elbow method to predict K

For each case we calculate **sum of squared Euclidian distances to the closest centroid**
Elbow method to predict K

For each case we calculate **sum of squared Euclidian distances to the closest centroid**

Elbow method for K-means: https://bl.ocks.org/rpgove/0060ff3b656618e9136b
Elbow method to predict K

Elbow method for K-means: https://bl.ocks.org/rpgove/0060ff3b656618e9136b

- K = 1: SSE = 35.42
- K = 2: SSE = 8.16
- K = 3: SSE = 5.33

Centroids and points for each K value.
Elbow method to predict K

K = 1

SSE: 35.42

K = 2

SSE: 8.16

K = 3

SSE: 5.33

Elbow method for K-means: https://bl.ocks.org/rpgove/0060ff3b656618e9136b
Elbow method to predict K

Elbow method for K-means: https://bl.ocks.org/rpgove/0060ff3b656618e9136b
Elbow method to predict K

For K = 1, SSE: 35.42
For K = 2, SSE: 8.16
For K = 3, SSE: 5.33

Elbow method for K-means: https://bl.ocks.org/rpgove/0060ff3b656618e9136b
Elbow method to predict K

SSE: 35.42

K = 1

SSE: 8.16

K = 2

SSE: 5.33

K = 3

Elbow method for K-means: https://bl.ocks.org/rpgove/0060ff3b656618e9136b
Any good way to choose $K$?

The rule of thumb is to choose $\sqrt{\frac{n}{2}}$ as $K$.

**Elbow method**: increase $K$ until it does not help to describe data better.
Any good way to choose $K$?

No, not really.

The rule of thumb is to choose $\sqrt{\frac{n}{2}}$ as $K$.

**Elbow method**: increase $K$ until it does not help to describe data better.
Hierarchical clustering

K-means

DBSCAN

Does not require K
Density-based spatial clustering of applications with noise (DBSCAN)
Two important parameters:

DBSCAN
Two important parameters:

- **epsilon**
- **Radius of the neighbourhood**, which is searched for each point

DBSCAN
Two important parameters:

- **epsilon**: Radius of the neighborhood, which is searched for each point.
- **minPoints**: Minimum number of points in the neighborhood to be in a cluster.
DBSCAN

Pick a random point
Pick a random point

Start

minPoints = 3

epsilon

DBSCAN
Pick a random point

Start

Search its **epsilon** neighbourhood

**epsilon**

minPoints = 3

DBSCAN
### DBSCAN

**Pick a random point**

If in **neighbourhood**

\[ \geq \text{minPoints} \]

points, this is **cluster**

- **minPoints = 3**
- **epsilon**

Start
If in **neighbourhood** >= **minPoints** points, this is **cluster** (including itself)

**minPoints** = 3

**epsillon**
If in *neighbourhood* >= *minPoints* points, this is *cluster* (including itself)

**DBSCAN**

\[ \text{minPoints} = 3 \]
DBSCAN

Move to the next point in the neighbourhood
Move to the next point in the neighbourhood

epsilon

minPoints = 3
Move to the next point in the neighbourhood

minPoints = 3

epsilon
Move to the next point in the neighbourhood.
**DBSCAN**

- **epsilon**
- **minPoints = 3**

**Move to the next point in the neighbourhood**
If no points left in the neighbourhood, pick another random point.
DBSCAN

$\minPoints = 3$

epsilon
DBSCAN

minPoints = 3

epsilon
DBSCAN

minPoints = 3

epsilon
This point does not have enough points, but it is reachable from the point that belongs to cluster.

minPoints = 3

epsilon
This point does not have enough points, but it is reachable from the point that belongs to cluster.
DBSCAN

minPoints = 3

epsilon

epsilon
DBSCAN

minPoints = 3

epsilon
Lonely points, are considered noise by DBSCAN.

minPoints = 3

epsilon
Very good visualisation: https://www.naftaliharris.com/blog/visualizing-dbscan-clustering/
Hierarchical clustering

Parameters: linkage type and the distance

K-means

Parameters: K and the distance

DBSCAN

Parameters: minPoints and epsilon
Hierarchical clustering
Parameters: linkage type and the distance

K-means
Parameters: K and the distance

DBSCAN
Parameters: minPoints and epsilon

Which of these three we should use and when?
Hierarchical clustering

Parameters: **linkage type** and the **distance**

K-means

Parameters: **K** and the **distance**

DBSCAN

Parameters: **minPoints** and **epsilon**
Hierarchical clustering

Using a **dendrogram** can extract **any** number of clusters

Parameters: **linkage type** and the **distance**

K-means

Parameters: **K** and the **distance**

**DBSCAN**

Parameters: **minPoints** and **epsilon**

**noise**

**centroids**
Hierarchical clustering

Hierarchical clustering is very slow

Using a **dendrogram** can extract **any** number of clusters

Parameters: **linkage type** and the **distance**

---

K-means

Parameters: **K** and the **distance**

---

DBSCAN

Parameters: **minPoints** and **epsilon**

---

Hierarchical clustering is very slow

---

Using a **dendrogram** can extract **any** number of clusters

Parameters: **linkage type** and the **distance**

---

K-means

Parameters: **K** and the **distance**

---

DBSCAN

Parameters: **minPoints** and **epsilon**
Hierarchical clustering

Parameters: linkage type and the distance

Using a **dendrogram** can extract **any number** of clusters.

Hierarchical clustering is **very slow** ($N^3$)

---

K-means

Parameters: $K$ and the distance

---

DBSCAN

Parameters: $\text{minPoints}$ and $\epsilon$

---

Noise

---

Centroids
Hierarchical clustering is very slow ($N^3$)

Using a dendrogram can extract any number of clusters.

Parameters: linkage type and the distance

K-means

Centroids

Parameters: $K$ and the distance

DBSCAN

Parameters: minPoints and epsilon
Hierarchical clustering is very slow ($N^3$)

Using a **dendrogram** can extract any number of clusters

**Parameters:** linkage type and the distance

Hierarchical clustering

K-means

You will have exactly **K** clusters

**Parameters:** K and the distance

DBSCAN

**Parameters:** minPoints and epsilon

**noise**

**centroids**
### Hierarchical clustering

- **Parameters:** linkage type and the distance

- **Pros:**
  - Using a **dendrogram** can extract any number of clusters.

- **Cons:**
  - Hierarchical clustering is **very slow** ($N^3$).

### K-means

- **Parameters:** $K$ and the distance

- **Pros:**
  - You will have exactly $K$ clusters.

- **Cons:**
  - You need to know $K$.

### DBSCAN

- **Parameters:** $\text{minPoints}$ and $\epsilon$

- **Pros:**
  - Noise detected.

- **Cons:**
  - Clustering can be challenging.
Hierarchical clustering is very slow ($N^3$).

Using a **dendrogram** can extract any number of clusters.

**Parameters:** linkage type and the distance

---

**K-means**

You will have exactly $K$ clusters.

Although **NP-hard**, it works surprisingly fast.

**Parameters:** $K$ and the distance

---

**DBSCAN**

You need to know $K$.

**Parameters:** minPoints and epsilon
**Hierarchical clustering**

- Parameters: linkage type and the distance
- Using a dendrogram can extract any number of clusters
- Hierarchical clustering is very slow ($N^3$)

**DBSCAN**

- Parameters: minPoints and epsilon
- Does not depend on $K$

**K-means**

- Parameters: $K$ and the distance
- You need to know $K$
- You will have exactly $K$ clusters
- Although NP-hard, it works surprisingly fast

**Hierarchical clustering**

- Parameters: linkage type and the distance
- Using a dendrogram can extract any number of clusters
- Hierarchical clustering is very slow ($N^3$)

**DBSCAN**

- Parameters: minPoints and epsilon
- Does not depend on $K$

**K-means**

- Parameters: $K$ and the distance
- You need to know $K$
- You will have exactly $K$ clusters
- Although NP-hard, it works surprisingly fast
References

• Machine Learning by Andrew Ng (https://www.coursera.org/learn/machine-learning)

• Curse of dimensionality: http://cleverowl.uk/2016/02/06/curse-of-dimensionality-explained/

• Visualisation of PCA: http://setosa.io/ev/principal-component-analysis/


• Stanford CS class: Convolutional Neural Networks for Visual Recognition by Andrej Karpathy (http://cs231n.github.io/)

• Data Mining Course by Jaak Vilo at University of Tartu (https://courses.cs.ut.ee/MTAT.03.183/2017_spring/uploads/Main/DM_05_Clustering.pdf)
What time is it now?
Kahoot!
That's all Folks!

BIIT

www.slideshare.net/DimaFishman